

Appl. No. 09/784,850
Amdt. Dated July 7, 2004
Reply to Office action of April 08, 2004
Attorney Docket No. P14282/040091-003
EUS/J/P/04-6147

Amendments to the Claims:

This listing of Claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Original) A method of processing data in a multi-processor environment, comprising the steps of:
 - building an application chain comprising at least one application to be performed on the data;
 - receiving the data;
 - adding program information to the received data for identifying a current application in the application chain to be performed on the data;
 - forwarding the data to an available processor in the multi-processor environment;
 - processing the data on the available processor using the current application identified by the added program information; and
 - updating the processed data with new program information identifying a next application in the application chain to be performed, if any, wherein the forwarding, processing and updating steps are repeated until the data is processed by all applications in the application chain.
2. (Original) The method of claim 1, wherein the data is received in the form of packets, each packet including both a header for storing the added and updated program information, and a payload for storing the received and processed data.
3. (Original) The method of claim 1, further comprising the step of forwarding code for the current application from at least one code server operating in the multi-processor environment to the available processor for use in processing the data if the code for the current application is not already installed on the available processor.

Appl. No. 09/784,850
Amdt. Dated July 7, 2004
Reply to Office action of April 08, 2004
Attorney Docket No. P14282/040091-003
EUS/JIP/04-6147

4. (Original) The method of claim 3, further comprising the step of removing the code for the current application from the available processor after the data has been processed.
5. (Original) The method of claim 1, further comprising the step of retrieving additional data needed for performing the application from an address location determined using the program information.
6. (Original) The method of claim 1, further comprising the step of storing at least one of the processed data, the program information, and an output state of the available processor at an address location determined using the program information after processing of the data by the current application is completed.
7. (Original) The method of claim 1, wherein the step of building an application chain comprises of the steps of forming a link for each of the at least one applications in the application chain, the link having information comprising:
 - a processor pool address identifying where in the multi-processor environment the available processor is to be located;
 - an application identifier for the application the link represents;
 - a first pointer identifying one of a previous link in the application chain and an entry point for the data;
 - a second pointer identifying one of a next link in the application and an exit point for the data;
 - a priority associated with the application; and
 - a link identifier indicating whether a next link in the application chain includes an application identifier.
8. (Original) The method of claim 7, wherein the program information added to the received data for identifying a current application in the application chain to be

Appl. No. 09/784,850
Amdt. Dated July 7, 2004
Reply to Office action of April 08, 2004
Attorney Docket No. P14282/040081-003
EUS/J/P/04-6147

performed on the data is the first pointer included in the link corresponding to the current application in the application chain.

9. (Original) The method of claim 7, wherein the processor pool address specifies a plurality of processors located within at least one of a communication network, a network node, a magazine of a network node, a media stream board, and a subset of a plurality of processors located on a media stream board.

10. (Original) The method of claim 7, further comprising the step of storing at least some of the link information in a connection table having a record for each link in the application chain, the record being indexed by the corresponding first pointer and including:

- the application identifier for the application the corresponding link represents;
- data information for the corresponding link;
- the priority associated with the application;
- the link identifier indicating whether the next link in the application chain includes an application identifier; and
- the corresponding second pointer.

11. (Original) The method of claim 10, wherein the data information comprises:
a data entry field specifying a number of data entries to be used by the current application;

- an address field indicating a starting address for a respective entry;
- a length field indicating a length in memory for a respective entry;
- a read bit indicating whether data should be read from the address of a respective entry and used by the current application when processing the data; and
- a write bit indicating whether data should be written to the address of a respective entry after execution of the current application.

Appl. No. 09/784,850
Amdt. Dated July 7, 2004
Reply to Office action of April 08, 2004
Attorney Docket No. P14282/040091-003
EUS/JP/04-6147

12. (Original) The method of claim 1, wherein certain applications are pre-installed on a plurality of processors operating in the multi-processor environment.
13. (Original) The method of claim 1, wherein all applications of a respective application chain are executed on a respective processor of the multi-processor environment.
14. (Original) The method of claim 1, wherein the multi-processor environment is a communication system.
15. (Original) The method of claim 1, wherein the applications are media stream applications.
16. (Original) An apparatus for processing data in a multi-processor environment, comprising:
 - logic that builds an application chain comprising at least one application to be performed on the data;
 - logic that receives the data;
 - logic that adds program information to the received data for identifying a current application in the application chain to be performed on the data;
 - logic that forwards the data to an available processor in the multi-processor environment;
 - logic that processes the data on the available processor using the current application identified by the added program information; and
 - logic that updates the processed data with new program information identifying a next application in the application chain to be performed, if any, wherein the forwarding, processing and updating steps are repeated until the data is processed by all applications in the application chain.

Appl. No. 09/784,850
Amdt. Dated July 7, 2004
Reply to Office action of April 08, 2004
Attorney Docket No. P14282/040091-003
EUS/J/P/04-6147

17. (Original) The apparatus of claim 16, wherein the data is received in the form of packets, each packet including both a header for storing the added and updated program information, and a payload for storing the received and processed data.

18. (Original) The apparatus of claim 16, further comprising logic that forwards code for the current application from at least one code server operating in the multi-processor environment to the available processor for use in processing the data if the code for the current application is not already installed on the available processor.

19. (Original) The apparatus of claim 18, further comprising logic that removes the code for the current application from the available processor after the data has been processed.

20. (Original) The apparatus of claim 16, further comprising logic that retrieves additional data needed for performing the application from an address location determined using the program information.

21. (Original) The apparatus of claim 16, further comprising logic that stores at least one of the processed data, the program information, and an output state of the available processor at an address location determined using the program information after processing of the data by the current application is completed.

22. (Original) The apparatus of claim 16, wherein the logic that builds an application chain comprises logic that forms a link for each of the at least one applications in the application chain, the link having information comprising:

- a processor pool address identifying where in the multi-processor environment the available processor is to be located;
- an application identifier for the application the link represents;
- a first pointer identifying one of a previous link in the application chain and an entry point for the data;

Appl. No. 09/784,850
Amdt. Dated July 7, 2004
Reply to Office action of April 08, 2004
Attorney Docket No. P14282/040091-003
EUS/J/P/04-6147

a second pointer identifying one of a next link in the application and an exit point for the data;

a priority associated with the application; and

a link identifier indicating whether a next link in the application chain includes an application identifier.

23. (Original) The apparatus of claim 22, wherein the program information added to the received data for identifying a current application in the application chain to be performed on the data is the first pointer included in the link corresponding to the current application in the application chain.

24. (Original) The apparatus of claim 22, wherein the processor pool address specifies a plurality of processors located within at least one of a communication network, a network node, a magazine of a network node, a media stream board, and a subset of a plurality of processors located on a media stream board.

25. (Original) The apparatus of claim 22, further comprising logic that stores at least some of the link information in a connection table having a record for each link in the application chain, the record being indexed by the corresponding first pointer and including:

the application identifier for the application the corresponding link represents;

data information for the corresponding link;

the priority associated with the application;

the link identifier indicating whether the next link in the application chain includes an application identifier; and
the corresponding second pointer.